Journal of Network Intelligence ©2024 ISSN 2414-8105
Taiwan Ubiquitous Information Volume **9**, Number **4**, November **2024**

Intelligent Quality Inspection of Customer Service Voice Based on Convolutional Neural Network and Transformer

Xiang-Na Li

Beijing FibrLink Communications Co., Ltd., Beijing 100054, P. R. China
wangkea_a2@163.com

Fang Xu

Beijing FibrLink Communications Co., Ltd., Beijing 100054, P. R. China
wangushen_dd@163.com

Xue-Dong Li*

Beijing FibrLink Communications Co., Ltd., Beijing 100054, P. R. China
g33669921@163.com

Bin Ma

Beijing FibrLink Communications Co., Ltd., Beijing 100054, P. R. China
lc12212002@163.com

Zhen-Xiang Pan

Beijing FibrLink Communications Co., Ltd., Beijing 100054, P. R. China
kx19981111@163.com

Shu-Feng Kou

Beijing FibrLink Communications Co., Ltd., Beijing 100054, P. R. China
Graduate School of Christian Studies
Baekseok University, Seoul 06695, South Korea
lvf0398@163.com

Zhao-Jun Yuan

State Grid Information & Telecommunication Branch, Beijing 100054, P. R. China
a363392022@163.com

*Corresponding author: Xue-Dong Li

Received January 10, 2024, revised April 15, 2024, accepted July 2, 2024.

Abstract. *Customer service voice quality inspection is a very important quality control link in hotline service operation. The traditional customer service voice QA system listens to the recordings against the assessment standard table manually, and mainly carries out sampling, which has low QA efficiency and small coverage. Therefore, it is proposed to use deep neural network model to calculate the text similarity, and by finding the deviation rate for quality control scoring, so as to greatly improve the efficiency and accuracy of quality inspection. First of all, data cleaning is first performed on the acquired raw text, and Chinese word splitting and de-duplication are carried out using Hanlp word splitting tool. Then, considering the importance of local and global features in natural language processing, this paper proposes a 2D U-Net deep learning model (2DTUnet) containing Transformer architecture. The Transformer module is used in the bridge for global feature modelling. Cascade connections from the encoder output after passing through the Transformer block are added in the decoder. Finally, the BM25 algorithm is introduced to measure sentence similarity and improve the weight calculation of links. The Word2Vec algorithm is used to train the word vectors and build the word vector model. After the processing of Word2Vec algorithm generates the semantic expansion matrix as the input of 2DTUnet. Finally, the Sigmoid activation function of the activation layer is used for the output to determine the probability of belonging to a certain class, and then the probability is used to judge the degree of text similarity. The experimental results of customer service voice quality control scoring show that the average deviation rate of text similarity of 2DTUnet model is 0.037, which is significantly smaller than that of the U-net model (0.058), which indicates that the 2DTUnet model can better satisfy the requirements of the enterprise for voice quality control services.*

Keywords: Speech quality control; deep learning model; text similarity; convolutional neural network; Transformer; word vector model

1. Introduction.

The voice customer service system records and stores a large number of recording files, in order to make strict supervision and objective evaluation of customer service quality, customer service centres often need to be equipped with a certain amount of quality inspectors through manual quality inspection to find out problems in the process of operation, so as to effectively improve the quality of service and the user's full degree [1, 2, 3]. Generally speaking, due to cost reasons, the manual sampling method usually has the problem of inefficiency in quality inspection. The ratio of customer service and QA personnel is 50:1, and the sampling is usually around 3% [4, 5]. Through intelligent Quality Inspection (QI), the QI sampling rate can reach 100%, and the QI personnel are mainly used to check the results of AI QI that are highly subjective, so that the QI efficiency can be greatly improved.

Intelligent quality inspection system needs to conduct real-time quality inspection of every sentence of customer service personnel according to real-time quality inspection scoring rules [6]. Real-time feedback and remind the customer service personnel to improve the quality of service. Need to support the creation of multiple quality control templates. Templates need to support the configuration of global rules, context rules, etc. [7, 8], in addition to supporting sensitive words (pre-set general sensitive words) model quality control. Intelligent QA system can extract valuable data information by analysing a large number of customer service recordings and form data reports. These reports can help enterprises gain insight into the performance of customer service teams, customer needs and complaints, and provide guidance for subsequent improvement and optimisation. When analysing voice recordings, the intelligent QI system can comprehensively consider multiple indicators, such as the accuracy of customer service pronunciation, clarity of expression, emotional control and other aspects [9]. Compared to manual QI, the intelligent system can assess customer service performance more comprehensively and objectively, and provide accurate ratings and feedback through algorithms. This can help companies understand the quality of customer service performance more accurately, and identify and correct problems in time to improve customer service quality.

According to relevant data, including more than 200 customer service centres will cover 75% of large and medium-sized enterprises, while the quality inspection personnel can only randomly sample less than 1% [10]. Obviously, the manual QI method cannot meet the demand. Therefore, the purpose of this work is to introduce the text similarity model and design an intelligent QI model for customer service voice based on deep learning technology, so as to improve the service quality and efficiency of the industry.

1.1. **Related Work.** The customer service voice intelligent quality control system mainly involves key technologies such as speech recognition and natural language processing [11], Chinese word segmentation [12], and calculating text similarity [13].

In terms of natural language processing, Goldberg [14] introduced machine learning and deep neural networks to improve the accuracy of speech recognition and natural language processing in practical applications. Li [15] introduced a convolutional network optimisation algorithm for speech recognition, and used the fractional order theory to deal with the Sigmoid function of the nodes in the convolutional neural network, which achieved the goal of reducing the training time and improving the training efficiency of the whole neural network. The purpose of reducing the training time and improving the training efficiency of the whole neural network is achieved.

Currently, there are three mainstream Chinese word segmentation algorithms: one is the algorithm based on word frequency statistics, which uses a large amount of corpus to train the word frequency of each word, and then cuts the text according to the word frequency; the second is the algorithm based on thesaurus, which uses the pre-compiled vocabulary list to perform the matching and segmentation; the third is the use of the conditional random field and maximum entropy model, and other statistical learning methods for word segmentation. Each of these three methods has its own strengths and weaknesses, and they are generally used in combination to improve the accuracy rate. Shu et al. [16] proposed a Chinese word segmentation method based on role annotation. In this method, the corpus is first labelled as a BMES system, then a model is trained by statistical machine learning, and finally the model is used to segment the unknown corpus. The advantage of this method is that it is simple and easy to use and has high accuracy, the disadvantage is that it needs a large amount of labelled corpus for training. Liu et al. [17] made some improvements on the basis of the minimum entropy model by inserting a virtual boundary character between the characters, and then using the method of minimizing cross entropy for training in order to achieve the segmentation of Chinese text. The advantage of this model is that it can flexibly deal with the complex and fuzzy characteristics of Chinese, but the disadvantage is that the computational volume is large.

Text similarity computation, as an applied basic technology of natural language processing, has attracted much attention. Chen [18] used semantic similarity to improve text topic analysis by LDA, and proposed that improving the reliability and performance of sentence similarity is the main research direction of text processing. Quan et al. [19] proposed the use of dependency grammars to compute sentence similarity, which considers that the similarity of utterances is due to the combination of word similarity and syntactic dependency. Tien and Labbé [20] analysed a variety of text similarity calculation methods and found that text similarity cannot only consider the literal meaning, but also explore the semantic similarity at a deep level. On the basis of text similarity comparison, Li et al. [21] proposed a short text semantic matching model with multi-granularity semantic crossover, and Mansoor et al. [22] proposed text semantic similarity computation based on deep Convolutional Neural Network (CNN). These methods and models have achieved good results in some domains, but there is room for improvement in specific scenario domains, such as customer service quality control systems.

1.2. Motivation and contribution.

Compared with CNN, U-Net is a deep learning network for image segmentation [23, 24], which can help us to distinguish different parts of an image. Then, the advantage of U-Net in the application of customer service speech mainly lies in its ability to better capture the correlation information between different time points when processing time series data. This means that in the QI process of speech text, U-Net can better understand the speech features at different points in time, thus improving the accuracy of recognition and QI. Although some 3D U-Net models are better for natural language processing, they all have the disadvantages of complex structure and difficult to be popularised, so they need to use an extremely high amount of memory, and the training produces an extremely large number of parameters, which makes the computation too complicated.

Therefore, to solve the above problem, this work proposes a 2D U-Net deep learning model incorporating the Transformer architecture [25] and employs it to implement an intelligent quality check for customer service speech. The main innovations and contributions of this work include:

(1) A new 2D Unet architecture (2DTUnet) is proposed for the traditional 3D U-Net architecture that suffers from complex structure and overly complicated computation, and the Transformer module is added to compensate for the lack of global contextual information modelling.

(2) 2DTUnet-based text similarity was introduced into the QA scoring model for automated scoring, which was used for score assessment of customer service quality. The performance of the QA automated scoring model on the historical dataset was evaluated using the deviation rate, and the results showed the effectiveness of the model.

2. Preprocessing of Natural Language Texts.

2.1. Chinese Participle Theory.

Natural language processing is an interdisciplinary discipline that integrates computer language and artificial intelligence as well as linguistics, and the ultimate goal of natural language processing is to let machines process and understand natural language and complete tasks with practical meaning. And language processing cannot be separated from the analysis of words to sentences and then to text, so Chinese participle, lexical annotation, named entity recognition, etc., are the preliminary areas of natural language processing, which lays the foundation for the implementation of the subsequent technology and practical applications.

Chinese word splitting is the process of splitting a piece of text into a series of words that are sequentially spliced together to equal the original text. There are two kinds of granularity in Chinese word splitting, coarse-grained word splitting and fine-grained word splitting. In practical applications, both coarse-grained and fine-grained participles have their own areas of expertise; coarse-grained participles are suitable for natural language processing applications, and fine-grained cuts are used for a long time in search engines.

Chinese participles, as the smallest unit that exists independently in natural language, are the beginning of natural language processing. If analysed from the perspective of computer linguistics, assuming that a sentence is transformed into a computable logical expression, the words in the sentence are computational symbols in the expression, some of which represent states or actions, and some of which are concepts of things. Chinese word-splitting algorithms are broadly divided into dictionary rule-based algorithms and machine learning-based algorithms.

The Hanlp lexer [26] used in this work is a set of toolkits that use cloud services to provide natural language processing services, which can provide web access interfaces and can be called through API interfaces or accessed and used directly on web pages. The Hanlp lexer does not require the lexer component toolkit to be downloaded to a local server, and the corpus domain used to train the lexer is better.

2.2. **Keyword Extraction Techniques.** Keyword extraction techniques are TextRank algorithm, PageRank, etc. The main idea of TextRank algorithm is to use a voting mechanism to vote the number of votes in favour of each word and its neighbouring window, and the weight of each word is determined by the number of votes. And PageRank uses a matrix iterative convergence method.

PageRank is a stochastic algorithm for web ranking [27], which hyperlinks from node V_i to node V_j as directed edges, initialising the weight $S(V_i)$ of each node to be 1, and updating the weight of each node in an iterative manner. The expression for updating the weights at each iteration is shown below:

$$S(V_i) = (1 - d) + d \times \sum_{V_j \in In(V_i)} \frac{1}{out(V_j)} S(V_j) \tag{1}$$

where d is a constant factor between $(0, 1)$, which is used in PageRank to simulate the probability of a user clicking on a link and thus jumping out of the current website; $In(V_i)$ denotes the set of nodes that are linked to the V_i; $Out(V_j)$ denotes the nodes that are linked from V_i. As you can see, it is not the case that the more links you have, the higher your PageRank will be. The more outbound links a site makes to other sites, the lower the weight of each outbound link.

The standard TextRank algorithm formula is based on the PageRank formula, and introduces the concept of edge weights as a way to represent the similarity of two sentences.

$$WS(V_i) = (1 - d) + d \times \sum_{V_j \in In(V_i)} \frac{\omega_{ji}}{\sum_{V_k \in Out(V_j)} \omega_{jk}} WS(V_j) \tag{2}$$

Assuming here that each word is considered as a sentence, the weight of all sentences or edges formed by words is 0, i.e., there is no intersection nor similarity between them.

2.3. **Semantic Vectors and Semantic Distances.** The most common first step in tasks related to natural language processing is to create a word list library and number each word sequentially, where each word is a very long vector. However, one of the biggest problems with this representation is that it does not capture the similarity and semantic distance between words well, especially in the field of deep learning where dimensional overloading is more likely to occur.

Therefore, each word is mapped into a k-dimensional real vector, and semantic similarity is determined by calculating the distances between words. Word2Vec [28] uses this form of distributional representation of word vectors, and is also a feature extraction technique in natural language processing. No matter how high the dimensionality of the original space in which the data is located and how sparse the distribution of the data is, after mapping it to a lower dimensional space, the distance between each other will be reduced and the similarity will be reflected.

Suppose there is a word vector of dimension k and the dictionary contains a total number of words n, so the word vector matrix is a matrix of $n \times k$, i.e., $W \in R^{nk}$, and then $W_i \in R^k$ denotes the word vector in the space of dimension k under the word vector

$\{t_{i1}, t_{i2}, \ldots, t_{ik}\}$. The final textual semantic similarity representation is computed using cosine similarity.

$$\mathrm{sim}_{w_i,w_j} = \cos(w_i, W_j) = \frac{W_i \cdot W_j}{\|W_i\|\|W_j\|} = \frac{\sum_{l=1}^{k}(t_{ik} \times t_{jk})}{\sqrt{\sum_{l=1}^{k} t_{ik}^2} \times \sqrt{\sum_{l=1}^{k} t_{jk}^2}} \tag{3}$$

where $0 \leq \cos(w_i, w_j) \leq 1$. The larger the value of sim_{w_i,w_j}, the greater the similarity between w_i and w_j.

At this stage, word vectors are trained on large-scale corpora mostly with neural network language models, so using them to represent text is more effective than traditional feature extraction methods and plays a role in solving the dimensionality problem of feature vectors.

3. CNN and Transformer based deep learning models.

3.1. Overall network structure design.

The most important aspect of processing natural language with the help of deep learning models is the accuracy of the model and the amount of computation. In recent years, a common method to increase the accuracy of deep learning models is to extend the 2D network to 3D.

In recent years, several effective deep learning models have used U-net networks. This structure always has the inherent locality of convolutional computation due to the fact that the sensory field of the convolutional kernel is ultimately limited, which causes it difficult for the U-net network to extract the global features of the brain tumour. In addition, although some 3D deep learning models are better for natural language processing, they all have the disadvantages of complex structure and difficult to be popularised, so they need to use an extremely high amount of memory, the training produces an extremely large number of parameters, and the computation is too large and too complex, so the configuration of the workstations is extremely demanding and not easy to implement. In order to solve the above problems, this paper proposes a 2D Transformer U-net (2DTUnet) based on the combination of CNN and Transformer. 2DTUnet's specific architecture is shown in Figure 1.

U-Net is a symmetric architecture, with the encoder part on the left and the decoder part on the right, constructed as a "U" shaped architecture. The encoder part reuses a pair of 3 × 3 convolutional layers to extract the feature maps. Downsampling is performed with the help of a 2×2 maximum pooling layer with a step size of 2, which doubles the number of feature channels after downsampling. Each convolution was followed by the use of a modified linear unit RELU decoder partially unused fully connected layers. The same iterative use of convolution and RELU. Up-sampling with the help of a 2×2 inverse convolution with a step size of 2 is used to recover feature map dimensionality and positional information and to fuse feature maps of different scales.

The jump-join section splices the feature maps from the output of each layer of the encoder with the feature maps from the decoder section after the up-sampling to compensate for the feature information lost during the down-sampling operation. This allows the decoder to incorporate high-resolution information from the encoding region during up-sampling, which further enhances the segmentation performance of the model. The ability of the Transformer to process the input feature information at several different locations simultaneously greatly increases the efficiency and scalability of the model.

3.2. Transformer Layer.

The input to the Transformer architecture is a one-dimensional sequence; first, the input feature vector is added before the encoder and decoder to give the feature vector a positional information, i.e., the positional encoding operation. The

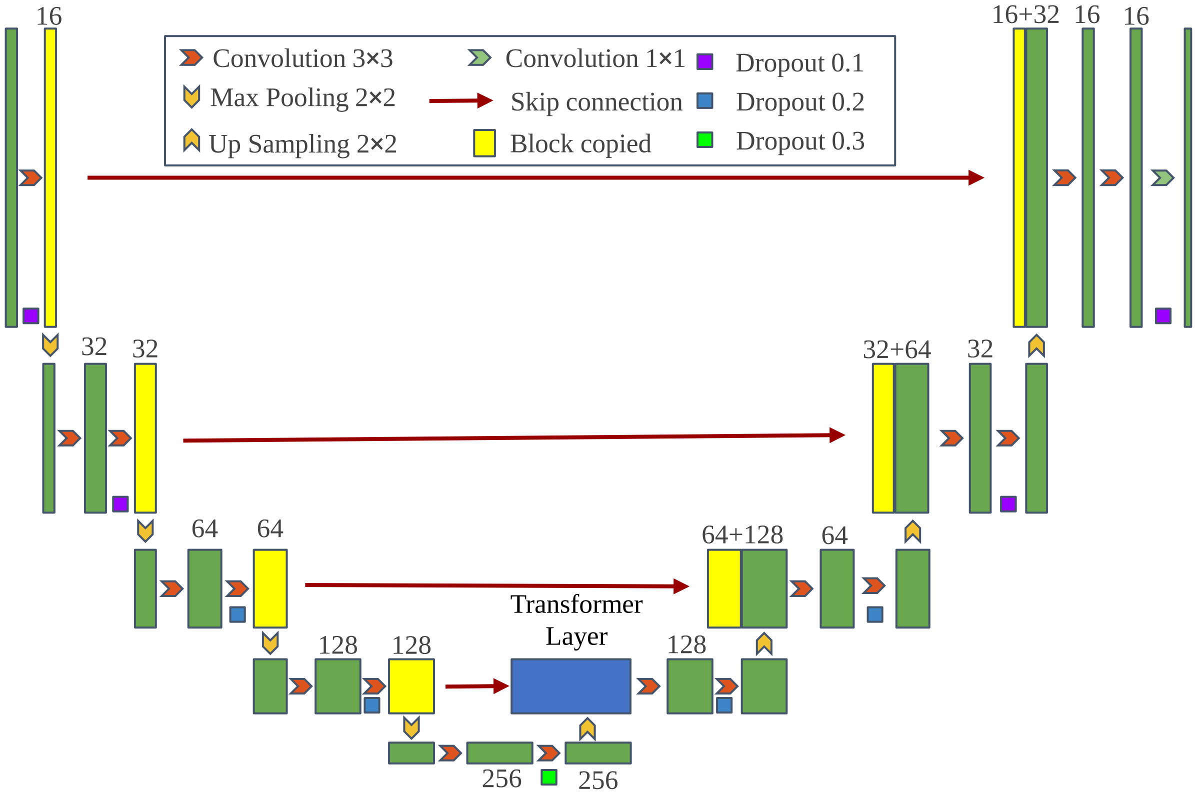

Figure 1. Structure of DTDUnet

encoder part is N stacked modules consisting of feed-forward neural network and normalization. The decoder part masks certain information so that it does not come into play when the parameters are updated. The output part generates the desired prediction probability information by linear transformation and Softmax operation. The structure of Transformer Layer is shown in Figure 2.

In the proposed 2DTUnet there is a need to input the pre-processed features into the Transformer Layer so as to extract the features thereby generating the interactive features, which is also known as the Transformer coding block. The Transformer coding block is mainly composed of four important parts, which are Layer Normalization (LN), Multi-HeadSelf-Attention, Drop Path, and Multi-Layer Perceptron (MLP) blocks. The expression of Transformer coding block is shown below:

$$z'_\ell = \mathrm{MSA}(\mathrm{LN}(z_{\ell-1})) + z_{\ell-1}, \quad z_\ell = \mathrm{MLP}(\mathrm{LN}(z'_\ell)) + z'_\ell \tag{4}$$

where $\ell = 1, \ldots, L$, L is the number of encoded block stacking; LN is mainly used for natural language processing tasks.

LN mainly deals with a single sample; batch normalization is to deal with all the samples of the same batch, from the calculation of the relationship between the two do not feel much difference, are subtracted and then divided by the standard deviation. In order to avoid the overfitting problem of the network, this paper chooses to use Drop Path in the Transformer Layer. Drop Path can be regarded as a special kind of Dropout, the difference is that it randomly discards sub-paths in the multi-branch structure of the deep learning model. Random samples in a batch will not pass through the backbone, but will be mapped directly through the branches in a constant manner. The GELU is the Gaussian Error Linear Unit. The first fully connected layer is followed by The first fully connected layer will change the dimension of the input to four times the original one, from $(N+1) \times D$ to $(N+1) \times 4D$, while the second fully connected layer will reduce the output back to $(N+1) \times D$.

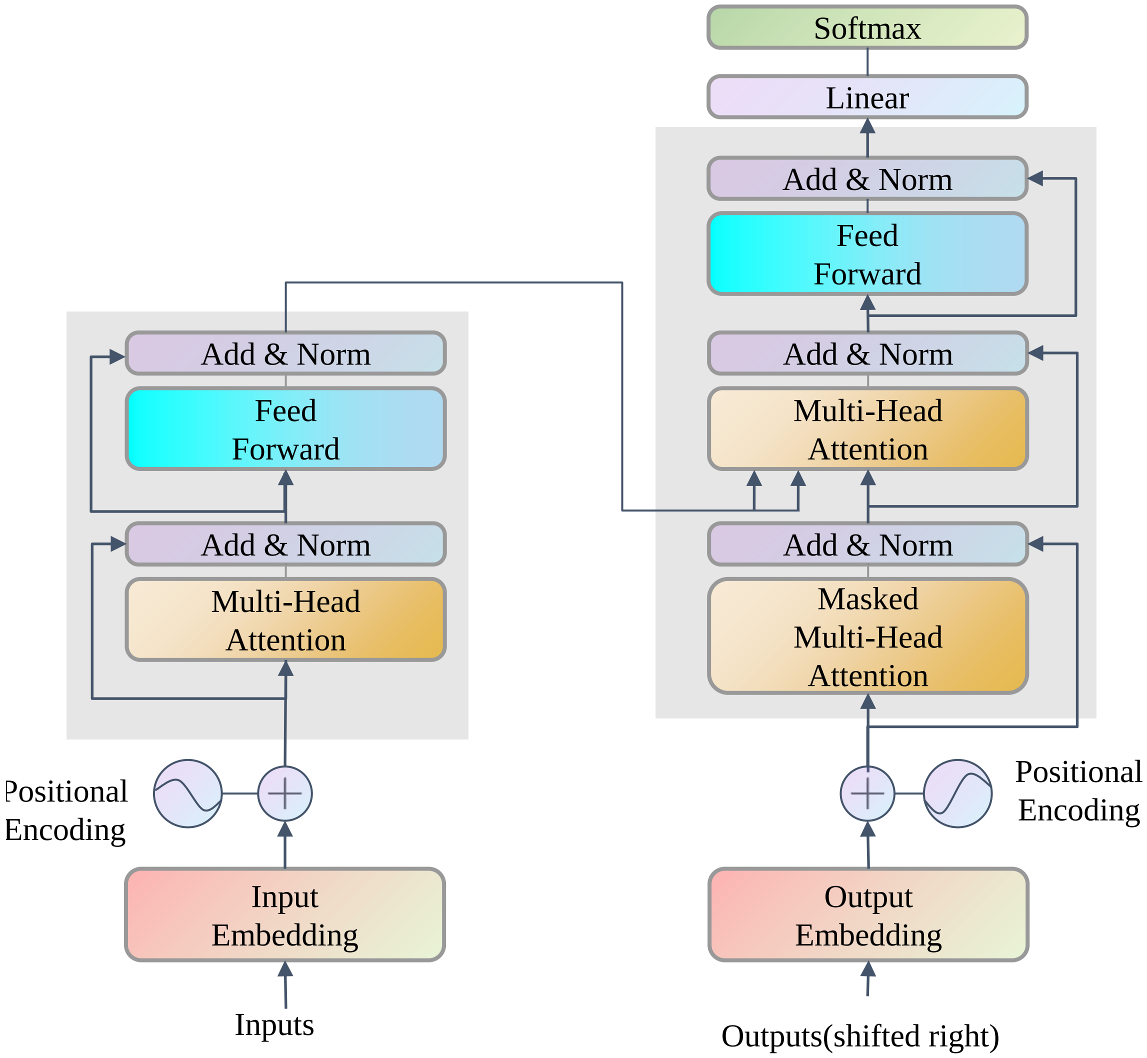

Figure 2. MMH-PSO algorithm for membrane structure

4. **2DTUnet based voice quality control model for customer service.**

4.1. **Key sentence extraction.** In the process of quality control, customer service dataset is large, colloquial and other characteristics, in this paper, the introduction of key sentence extraction, the purpose of which is to extract the central content from the text of a large length.

The traditional key sentence extraction is using PageRank. However, the traditional PageRank does not work on sentence granularity due to the low probability of having the same two sentences in the same text. In order to apply PageRank to sentence granularity, this paper introduces the BM25 algorithm to measure sentence similarity and improve the weight calculation of links. In this way the links between the central sentence of the window and the neighbouring sentences become strong or weak, and similar sentences will get higher votes. However, the key sentences used to express the central meaning in the text tend to have higher similarity with other sentences explaining the description, which is in line with the BM25 algorithm.

In this module, the BM25 algorithm is used to measure the degree of association of multiple words with the text. Here we define Q as a query statement which consists of keywords $q_1, q_2, \ldots, q_n$. D is a retrieved text, and the similarity between them is defined by the BM25 measure as shown below:

$$BM25(D,Q) = \sum_{i=1}^{n} IDF(q_i) \cdot \frac{TF(q_i,D) \cdot (k_1+1)}{TF(q_i,D) + k_1 \cdot \left(1 - b + b \cdot \frac{|D|}{avgDL}\right)} \tag{5}$$

where k_1 and b are two constants; $avgDL$ is the average length of all texts; $TF(q_i, D)$ stands for how often q_i occurs in d; $IDF(q_i)$ is the reciprocal of DF; The inverse of $DF(q_i)$ represents how many texts contain q_i.

Comparing the definition formula of TF-IDF with BM25, it is easy to see that BM25 means weighted sum of IDFs of all words in the query statement, and the two constant parameters and TF can be regarded as the parameters to adjust the weights of IDFs. The larger k_1 is, the greater the positive impact of TF on the document score. The larger b is, the greater the negative impact of document length on the score. When $k_1 = b = 0$, BM25 is completely equivalent to the sum of IDFs of all words. In TF-IDF, when IDF is fixed, the score is proportional to TF, so that long documents are inherently more advantageous, and therefore, BM25 is significantly more refined in this way of handling TF.

After using the BM25 algorithm, by considering a sentence as a query statement and the neighbouring sentences as documents to be queried, the similarity between them is obtained. This similarity is used as the weight of the links in TextRank.

$$WS\left(V_i\right) = (1-d) + d \times \sum_{V_j \in \mathrm{In}(V_i)} \frac{BM25\left(V_i, V_j\right)}{V_k \in \mathit{out}\left(V_j\right) BM25\left(V_k, V_j\right)} WS\left(V_j\right) \tag{6}$$

where $WS(V_i)$ is the score of the i-th sentence in the document, repeated iteration of the expression a number of times to get the final score, sorted output of the first N that is, to get the key sentence. In addition, because the number of sentences in the document is much smaller than the number of words, and the sentences are almost not repeated, so usually no longer take the window, but that all the sentences are adjacent.

4.2. **Word Vector Building Based on Word2Vec Algorithm.** In this work, Word2Vec algorithm is used to train word vectors and build word vector models. After the processing of Word2Vec algorithm generates the semantic expansion matrix as the input of 2DTUnet.

The first step is to select the activation function of the hidden layer, here we identify tanh as the activation function in the hidden layer as in Equation (7). The activation function of the hidden layer is a form of realisation of one component on each vector. Next, a vector $y_w = (y_{w,1}, y_{w,2}, \ldots, y_{w,N})$ of length N is computed. However, the value range of the y_w component result cannot be between 0 and 1, and the word context form cannot be expressed as Context(w). The last step is to use softmax function to calculate the value of $p(w|\text{Context}(w))$. Then the probability that the next word should be exactly the i-th word in the dictionary D.

$$Z_w = \tanh(Wx_{y_w} + p) \tag{7}$$

$$y_w = UZ_w + q \tag{8}$$

$$p(w|\text{Context}(w)) = \text{Softmax}(y_w) \tag{9}$$

where i denotes the index of this w in the dictionary D. Softmax is a case extension of the Context(w) function for multiclassification. Context(w) can be used for binary classification. For multiclassification problems, the most commonly used function for deep learning is Softmax.

Given the characteristics of the QI dataset, the CBOW model was chosen in the final study to reduce the complexity of the approximation method. CBOW is a model similar to the forward NNLM. The basic principle of the CBOW model is to use the word vectors of the surrounding words of the current word to be predicted as the input layer and the output of the output layer is the word vector of the current word. The learning objective of this model is to maximise the log-likelihood function.

$$\delta = \sum_{w \in C} \log p(w|\text{Context}(w)) \tag{10}$$

where w is any word in the corpus.

The vocabulary of the corpus is fixed and the output layer outputs the most likely w. The process can also be viewed as a multi-classification process. For multiclassification neural network models, the most common approach is Softmax regression.

$$h_0(x^{(i)}) = \begin{bmatrix} p\left(y^{(i)} = 1|x^{(i)}; \theta\right) \\ p\left(y^{(i)} = 2|x^{(i)}; \theta\right) \\ \vdots \\ p\left(y^{(i)} = k|x^{(i)}; \theta\right) \end{bmatrix} = \frac{1}{\sum_{j=1}^{k} e^{\theta_j^T x^{(i)}}} \begin{bmatrix} e^{\theta_1^T x^{(i)}} \\ e^{\theta_2^T x^{(i)}} \\ \vdots \\ e^{\theta_k^T x^{(i)}} \end{bmatrix} \tag{11}$$

where $V \in \mathbb{R}^{|h| \times v}$ denotes the weight matrix of the neural network model from the input layer to the hidden layer; $U \in \mathbb{R}^{h \times m}$ denotes the weight matrix from the hidden layer to the input layer; n is the dimension of the word vectors; V and U are referred to as the input word vector matrix and the output word vector matrix, respectively; V_i denotes the input word vector of the i-th word in V; U_j denotes the output word vector of the j-th word in U.

In order to make the CBOW model prediction more accurate and want the estimation of $\hat{y} \in \mathbb{R}^{|V|}$ and the real probability distribution $y \in \mathbb{R}^{|V|}$ to be as similar as possible, so the present work uses the cross-entropy as the loss function, and the stochastic gradient descent algorithm is used to optimise the two parameter matrices V and U. Where the cross entropy loss function is defined as follows.

$$H(\hat{y}, y) = -\sum_{j=1}^{|V|} y_j \log(\hat{y_j}) \tag{12}$$

4.3. **2DTUnet-based text similarity computation.** Suppose m words in dictionary A are constructed into a word vector matrix $V \in \mathbb{R}^{m \times d}$, where $\text{vec}(c_j) \in \mathbb{R}^d$ denotes the word vector of the j-th word c_j in the d-dimensional space.

Given a text $D = \{c_1, c_2, \cdots, c_n\}$, where the text contains n words. Next, it will go through preprocessing such as word splitting and de-duplication. The preprocessed, semantically expanded text D is expanded and completed as DE, and the text semantic expansion matrix X is obtained. The text preprocessing process has been given in the previous section. First judge whether each word exists in the dictionary A, if it exists, add it to the vector, and add its corresponding word vector to the matrix X, otherwise ignore it.

When sliding the convolution kernel along the entire width and height of the input, it will have a 2D activation map, the feature map, which represents the response of the input to the convolution kernel at each spatial location. The U-net network typically produces a stronger response to literal meanings in the text, and we consider that this part of the extraction is mainly the low-level features. The Transformer layer tends to produce a strong response to a target that has a clear semantic meaning. The U-net network is to capture word n-gram contextual features through a sliding window. When $n = 1$, it is a context-independent language model. Here we generally consider a context-relative model when $n \geq 2$. The goal of general n-gram language model optimisation is maximum Log likelihood.

$$F = \max \sum_{t=1}^{T} \log p_m \left(w_t \mid w_{t-n+1}, w_{t-n+2}, \cdots, w_{t-1}\right) \tag{13}$$

The pooling layer is implemented to find and combine words with significant n-gram features to form an utterance-level feature vector of a certain length. Therefore, it is necessary to select the neuron activation value that has the largest activation value for each pooled region in the feature graph.

$$C_{\max} = \max \left(C_i\right) \tag{14}$$

where C_i is the feature map formed by the convolutional layer after the convolution operation on the extended semantic matrix. The final output of the pooling layer is provided by the maximum value of each feature map.

Feature vectors are generated at the utterance level by merging the two vectors generated in the vertical direction between the fully connected layers through the Merge layer. Non-linear variations are used to perform the extraction of high-level semantic representations.

$$y = \tan \left(W_s \cdot v\right) \tag{15}$$

where v denotes the global feature vector, W_s denotes the semantic projection matrix, and y denotes the potential semantic space vector.

Eventually, the high-level semantics generated by the fully connected layer will be output towards the probability of belonging to a certain class through the Sigmoid activation function of the activation layer, which will be used to determine the similarity between the texts. The output value is a floating-point number between 0 and 1, which indicates the similarity between the texts T_1 and T_2. The larger the output value, the more similar the two texts are. The final semantic similarity is attributed to the similarity value, which is used for similarity judgement. The closer the output value is to 1, the greater the semantic similarity and vice versa.

5. **Experimental results and analyses.**

5.1. **Training parameters and evaluation metrics.** After completing the data preprocessing, the CBOW model is used for word vector model training. For word vector training, 5308 original speech datasets are selected. The speech dataset comes from the historical data of customer voice services in the State Grid ICT Industry Group with a period of 4 months. The format of all speech sequences is MP4. The training parameters of Word2Vec model and 2DTUnet model are shown in Table 1.

The concept of deviation rate is introduced to quantify the performance of the two text similarities on the historical dataset. The score is measured according to the text similarity, and the deviation rate is calculated from the model QI score and the manual QI in previous years. The customer service voice intelligent quality inspection system is to automate the scoring of customer service quality, and the scoring model is calculated using the text similarity value combined with the quality inspection standard. The deviation rate is calculated as shown below:

$$D = \frac{|\text{score}_{\text{human}} - \text{score}_{\text{model}}|}{\text{score}} \tag{16}$$

$$D_{\text{avg}} = \frac{1}{n}\sum_{i=0}^{n} \frac{|\text{score}_{\text{human}} - \text{score}_{\text{model}}|}{\text{score}} \tag{17}$$

where $\text{score}_{\text{human}}$ denotes the score given as a result of manual scoring, $\text{score}_{\text{model}}$ denotes the score given as a result of model scoring, and score denotes the full score of the system. D is the deviation rate, which indicates the range of deviation between the automatic quality control score and the manual quality control score in a single speech text. D_{avg} is the average deviation rate, which is the average of the deviation rates of the automatic quality control scores and the manual scores using the text similarity model in all speech samples.

Table 1. Model training parameters

Parameters	**Set up**
Training window radius	8
Vocab_size	200
Sampling threshold	0.0001
Number of processes	4
Minimum frequency	5
Filter	128
Filter_window	3
Dropout	0.2
Iterations	10

5.2. **Model performance comparison.** In order to verify the effectiveness of the proposed 2DTUnet model in the customer service voice QI system, two text similarity calculations are used as the QI scoring models respectively to automatically score 500 randomly selected historical voice samples.

The first uses the U-net model, while the second uses the 2DTUnet model. The inputs of both models are processed by Word2Vec algorithm to generate semantic expansion matrices. The distribution results of the computed deviation rates for the two similarity QA scoring models are shown in Table 2. The comparison results of the minimum, average and maximum deviation rates are shown in Table 3.

Comparison of Tables 2 and 3 shows that the deviation rate of the U-net-based similarity scoring model is mainly in the range of 0.05-0.06, while that of the 2DTUnet-based similarity scoring model is mainly in the range of 0.03-0.04. Therefore, compared with the manual QI scores of the past years, the errors of the automatic scoring of the two text similarity computation models mentioned above are in the acceptable range. In addition, comparing the two models shows that the average deviation rate of text similarity of the 2DTUnet model is 0.037, which is significantly smaller than that of the U-net model

Table 2. Deviation rate distribution results

Deviation rate	**U-net**	**2DTUnet**
0.01-0.02	102	106
0.03-0.04	129	208
0.05-0.06	213	115
0.07-0.08	35	34
0.09-0.10	16	23
0.10 or more	5	14

Table 3. Deviation rate statistics

Deviation rate	**U-net**	**2DTUnet**
$D_{\min}$	0.01	0.01
D_{avg}	0.058	0.037
$D_{\max}$	0.13	0.112

(0.058), which indicates that the former has a higher precision and accuracy of the similarity value, so it is practicable to apply the text similarity model based on the 2DTUnet to the online customer service voice intelligent quality inspection system.

6. **Conclusion.** In this work, a 2DTUnet-based intelligent quality control model for customer service speech is proposed. The Transformer module is added to 2D U-Net for bridging the gap of global context information modelling. Word2Vec algorithm is used to train word vectors and build word vector models. The semantic expansion matrix is processed by Word2Vec algorithm to generate semantic expansion matrix as the input of 2DTUnet. The text similarity based on 2DTUnet was introduced into the QI scoring model for automated scoring, which was used for score assessment of customer service quality. The performance of the QA automated scoring model on the historical dataset was evaluated using the deviation rate. The experimental results show that the deviation rate of the 2DTUnet-based similarity scoring model is mainly in the range of 0.03-0.04. Compared with the manual QA scoring in the past years, the model error is in the acceptable range. However, the deepening of network depth is often accompanied by problems such as gradient vanishing and explosion. Therefore, subsequent studies will try to introduce stacked residual blocks to improve the learning ability of certain shallow layers.

REFERENCES

[1] C. F. Lam and D. M. Mayer, "When do employees speak up for their customers? A model of voice in a customer service context," *Personnel Psychology*, vol. 67, no. 3, pp. 637-666, 2014.

[2] R. Lacey, "How customer voice contributes to stronger service provider relationships," *Journal of Services Marketing*, vol. 26, no. 2, pp. 137-144, 2012.

[3] L. Lehto, L. Laaksonen, E. Vilkman, and P. Alku, "Changes in objective acoustic measurements and subjective voice complaints in call center customer-service advisors during one working day," *Journal of Voice*, vol. 22, no. 2, pp. 164-177, 2008.

[4] L. Wang, N. Huang, Y. Hong, L. Liu, X. Guo, and G. Chen, "Voice-based AI in call center customer service: A natural field experiment," *Production and Operations Management*, vol. 32, no. 4, pp. 1002-1018, 2023.

[5] D. Buhalis and I. Moldavska, "Voice assistants in hospitality: using artificial intelligence for customer service," *Journal of Hospitality and Tourism Technology*, vol. 13, no. 3, pp. 386-403, 2022.

[6] C. Chow-Chua and R. Komaran, "Managing service quality by combining voice of the service provider and voice of their customers," *Managing Service Quality: An International Journal*, vol. 12, no. 2, pp. 77-86, 2002.

[7] L. L. Bove and N. L. Robertson, "Exploring the role of relationship variables in predicting customer voice to a service worker," *Journal of Retailing and Consumer Services*, vol. 12, no. 2, pp. 83-97, 2005.

[8] A. Burgers, K. de Ruyter, C. Keen, and S. Streukens, "Customer expectation dimensions of voice-to-voice service encounters: a scale-development study," *International Journal of Service Industry Management*, vol. 11, no. 2, pp. 142-161, 2000.

[9] S. P. Xu, K. Wang, M. R. Hassan, M. M. Hassan, and C.-M. Chen, "An Interpretive Perspective: Adversarial Trojaning Attack on Neural-Architecture-Search Enabled Edge AI Systems," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 1, pp. 503-510, 2023.

[10] C. C. Aguwa, L. Monplaisir, and O. Turgut, "Voice of the customer: Customer satisfaction ratio based analysis," *Expert Systems with Applications*, vol. 39, no. 11, pp. 10112-10119, 2012.

[11] D. Khurana, A. Koli, K. Khatter, and S. Singh, "Natural language processing: State of the art, current trends and challenges," *Multimedia Tools and Applications*, vol. 82, no. 3, pp. 3713-3744, 2023.

[12] J. Hirschberg and C. D. Manning, "Advances in natural language processing," *Science*, vol. 349, no. 6245, pp. 261-266, 2015.

[13] A. Galassi, M. Lippi, and P. Torroni, "Attention in natural language processing," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 10, pp. 4291-4308, 2020.

[14] Y. Goldberg, "A primer on neural network models for natural language processing," *Journal of Artificial Intelligence Research*, vol. 57, pp. 345-420, 2016.

[15] H. Li, "Deep learning for natural language processing: advantages and challenges," *National Science Review*, vol. 5, no. 1, pp. 24-26, 2018.

[16] X. Shu, J. Wang, X. Shen, and A. Qu, "Word segmentation in Chinese language processing," *Statistics and its Interface*, vol. 10, no. 2, pp. 165-173, 2017.

[17] J. Liu, F. Wu, C. Wu, Y. Huang, and X. Xie, "Neural Chinese word segmentation with dictionary," *Neurocomputing*, vol. 338, pp. 46-54, 2019.

[18] L.-C. Chen, "An effective LDA-based time topic model to improve blog search performance," *Information Processing & Management*, vol. 53, no. 6, pp. 1299-1319, 2017.

[19] Z. Quan, Z.-J. Wang, Y. Le, B. Yao, K. Li, and J. Yin, "An efficient framework for sentence similarity modeling," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 4, pp. 853-865, 2019.

[20] N. M. Tien and C. Labbé, "Detecting automatically generated sentences with grammatical structure similarity," *Scientometrics*, vol. 116, no. 2, pp. 1247-1271, 2018.

[21] L. Li, M. Kong, D. Li, and D. Zhou, "A multi-granularity semantic space learning approach for cross-lingual open domain question answering," *World Wide Web*, vol. 24, no. 4, pp. 1065-1088, 2021.

[22] M. Mansoor, Z. Ur Rehman, M. Shaheen, M. A. Khan, and M. Habib, "Deep learning based semantic similarity detection using text data," *Information Technology and Control*, vol. 49, no. 4, pp. 495-510, 2020.

[23] N. Siddique, S. Paheding, C. P. Elkin, and V. Devabhaktuni, "U-net and its variants for medical image segmentation: A review of theory and applications," *IEEE Access*, vol. 9, pp. 82031-82057, 2021.

[24] M. Z. Alom, C. Yakopcic, M. Hasan, T. M. Taha, and V. K. Asari, "Recurrent residual U-Net for medical image segmentation," *Journal of Medical Imaging*, vol. 6, no. 1, pp. 014006-014006, 2019.

[25] K. Han, A. Xiao, E. Wu, J. Guo, C. Xu, and Y. Wang, "Transformer in transformer," *Advances in Neural Information Processing Systems*, vol. 34, pp. 15908-15919, 2021.

[26] M. Lei, B. Wen, J. Gan, and J. Wang, "Clustering algorithm of ethnic cultural resources based on spark," *International Journal of Performability Engineering*, vol. 15, no. 3, pp. 756, 2019.

[27] P. Zhang, T. Wang, and J. Yan, "PageRank centrality and algorithms for weighted, directed networks," *Physica A: Statistical Mechanics and its Applications*, vol. 586, pp. 126438, 2022.

[28] J.-J. Zhu and Z. J. Ren, "The evolution of research in resources, conservation & recycling revealed by Word2vec-enhanced data mining," *Resources, Conservation and Recycling*, vol. 190, pp. 106876, 2023.